Clustering Algorithms

k-Means and Hierarchical Clustering

Md Amanullah
Matrikelnummer: 5466324
CS2356

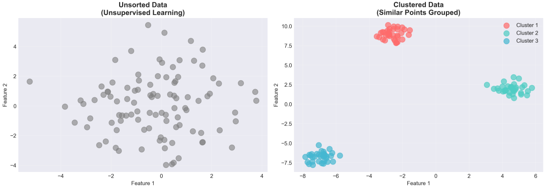
Machine Learning
Supervisor: Herr Cemic

January 13, 2026

Table of Contents

What is Clustering?

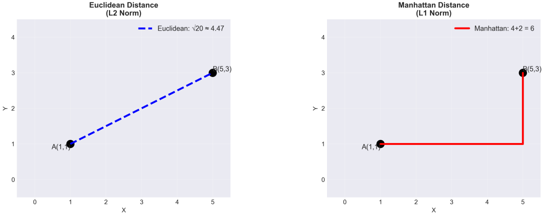
- **Unsupervised Learning**: No predefined labels
- **Objective**: Group similar data points
- **Core Principle**: Maximize intra-cluster similarity, minimize inter-cluster similarity
- **Applications**:
  - Customer segmentation
  - Image compression
  - Gene analysis
  - Document organization
  - Anomaly detection



Distance Metrics and Similarity

Euclidean Distance:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$



Intra-Cluster Distance:

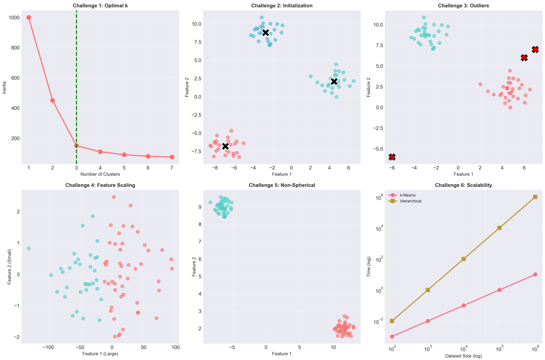
- Distance within cluster
- Should be **minimized**
- Indicates compactness

Inter-Cluster Distance:

- Distance between clusters
- Should be **maximized**
- Indicates separation

Clustering Challenges

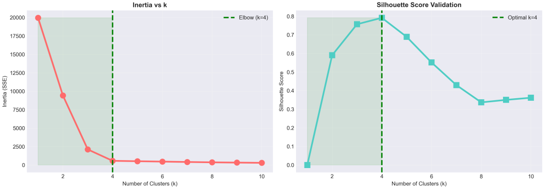
- **Optimal k**: How many clusters?
- **Initialization**: Different starting points
- **Scalability**: Large datasets
- **Outliers**: Noise distortion
- **Feature Scaling**: Normalization needed
- **Cluster Shapes**: Spherical assumption



Determining Optimal k: The Elbow Method

The Logic:

- As k increases, Inertia decreases
- Look for the "Elbow" point
- Point of diminishing returns
- Validate with Silhouette Score



Inertia Formula:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

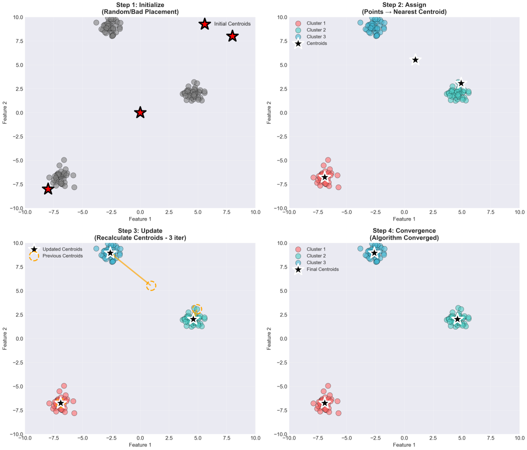
k-Means: Algorithm Overview

Core Concept: Partition data into k clusters through iterative centroid optimization

Algorithm Steps:

1. **Initialize**: Random k centroids
2. **Assign**: Points to nearest centroid
3. **Update**: Recalculate centroids
4. **Repeat**: Until convergence

Convergence: Centroids stabilize



k-Means++: Smart Initialization

The Problem with Random Initialization:

- Standard k-Means can converge to poor local minima
- Random centroid placement may require many iterations
- Results can be inconsistent across runs

The k-Means++ Solution:

1. Choose first centroid uniformly at random from data points
2. For each remaining centroid, choose a point with probability proportional to $D(x)^2$
3. Where $D(x)$ is the distance to the nearest existing centroid

Benefits:

- **Faster convergence**: Typically 2-3x fewer iterations
- **Better results**: Avoids poor local minima
- **Default in Scikit-Learn**: Used automatically in modern implementations

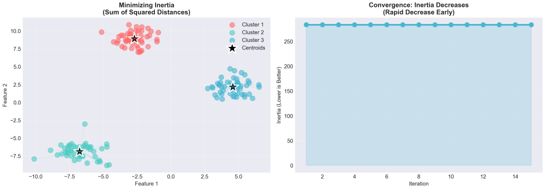
k-Means: Mathematical Foundation

Objective Function (Inertia):

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Where:

- C_i = Set of points in cluster i
- μ_i = Centroid (mean) of cluster i
- k = Number of clusters
- $\|x - \mu_i\|^2$ = Squared Euclidean distance

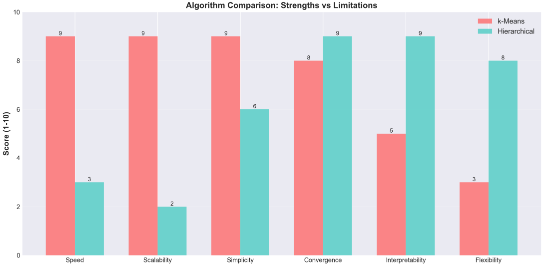


Centroid Update Rule:

$$\mu_i^{(t+1)} = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

k-Means: Strengths and Limitations

- **Strengths**:
  - Fast and efficient
  - Simple to implement
  - Scales well
  - Guaranteed convergence
  - Widely available
- **Limitations**:
  - Requires predefined k
  - Initialization sensitive
  - Local minima
  - Outlier sensitive
  - Assumes spherical clusters
  - Needs normalization



k-Means: Computational Complexity

Time Complexity:

$$O(n \cdot k \cdot d \cdot i)$$

Where:

- n = Number of data points
- k = Number of clusters
- d = Number of dimensions
- i = Number of iterations (typically 10-20)

Space Complexity: $O(n \cdot d + k \cdot d)$

Practical Note: Usually converges in 10-20 iterations, making it very efficient

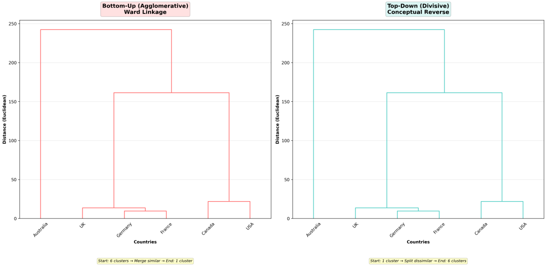
Hierarchical Clustering: Two Approaches

Agglomerative (Bottom-Up):

- Start with individual points
- Merge closest clusters
- Build tree from bottom up
- Most common approach
- Uses linkage methods

Divisive (Top-Down):

- Start with all points together
- Recursively split clusters
- Build tree from top down
- Less commonly used
- More expensive



Output: Dendrogram (tree structure)

Hierarchical Clustering: Linkage Methods

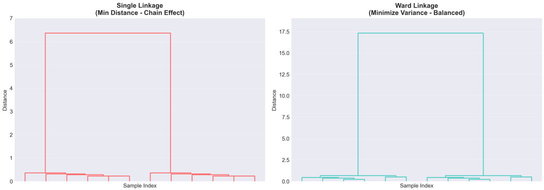
How to Measure Distance Between Clusters?

Single Linkage:

- Uses minimum distance
- Creates chain-like clusters
- Sensitive to outliers

Ward Linkage:

- Minimizes within-cluster variance
- Creates balanced clusters
- Often produces best results
- Recommended in practice



Hierarchical Clustering: Strengths and Limitations

Strengths:

- No predefined k needed
- Dendrogram interpretable
- Flexible selection
- Deterministic results
- Reveals hierarchy

Limitations:

- Expensive $O(n^2)$
- Cannot undo merges
- Linkage sensitive
- Poor scalability
- $O(n^2)$ memory
- Outlier sensitive

Hierarchical Clustering: Computational Complexity

Time Complexity:

$$O(n^2 \log n) \text{ to } O(n^3)$$

Depends on linkage method and implementation

Space Complexity: $O(n^2)$ for distance matrix

Practical Limitations:

- Suitable for datasets with fewer than 10,000 points
- Requires storing complete distance matrix in memory
- Not recommended for real-time applications
- Better suited for exploratory data analysis

k-Means vs. Hierarchical Clustering

Criterion	k-Means	Hier.
Time	O(nkdi)	$O(n^2 \log n)$
Space	O(nd+kd)	$O(n^2)$
Predefined k	Yes	No
Dendrogram	No	Yes
Scalability	High	Low
Interpret.	Medium	High
Deterministic	No	Yes

Key Trade-off: Speed vs. Interpretability

- k-Means: Fast but less interpretable
- Hierarchical: Slower but more interpretable

When to Use Each Algorithm

Use k-Means When:

- You have a large dataset (more than 10,000 points)
- You know the desired number of clusters
- Speed and efficiency are priorities
- You need real-time clustering
- Computational resources are limited

Use Hierarchical Clustering When:

- You want to explore cluster structure
- The number of clusters is unknown
- You need interpretable results
- You have a small to medium dataset
- Understanding relationships is important

Practical Applications

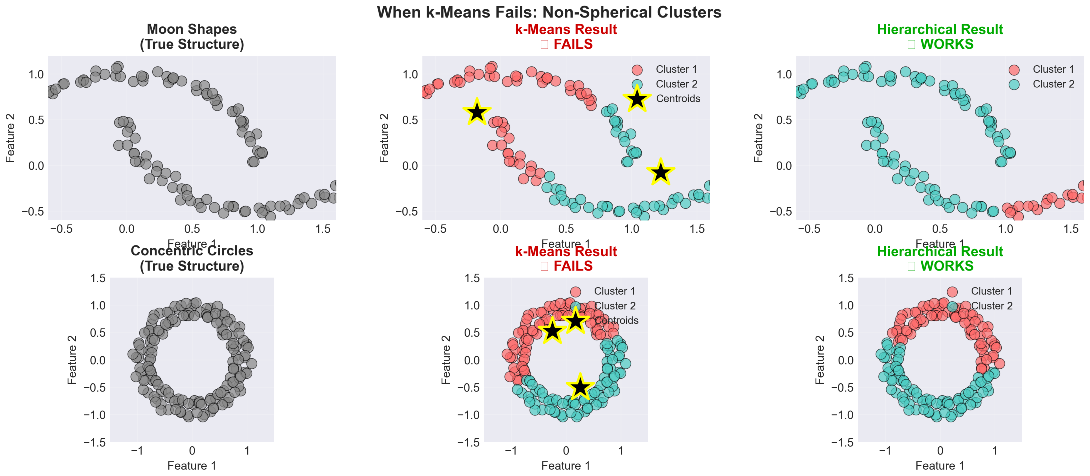
k-Means Applications:

- Customer segmentation for marketing campaigns
- Image compression and color quantization
- Document clustering and topic modeling
- Recommendation systems
- Network traffic anomaly detection

Hierarchical Clustering Applications:

- Gene sequence analysis and phylogenetics
- Document classification and organization
- Exploratory data analysis
- Taxonomy and hierarchy creation
- Social network community detection

When k-Means Fails: Non-Spherical Clusters



Lesson: Always visualize data first. Consider alternatives like DBSCAN for non-spherical clusters.

Key Discussion Question

How would you select the most appropriate clustering algorithm for a specific real-world application?

Critical Considerations:

- Dataset size and dimensionality
- Whether the number of clusters is known
- Trade-off between speed and interpretability
- Expected cluster shapes and sizes
- Available computational resources
- Need for hierarchical structure understanding

Summary and Key Takeaways

- **Clustering** is a fundamental unsupervised learning technique
- **k-Means**: Fast, efficient, scalable, but requires predefined k
- **Hierarchical**: Flexible, interpretable, but computationally expensive
- **Algorithm Selection** depends on specific application requirements
- **Both Methods** are valuable and often complementary
- **Data Preprocessing** (scaling, normalization) is crucial
- **Validation Metrics** (silhouette score) assess quality
- **Hybrid Approaches**: Use hierarchical for exploration, then k-Means for final clustering

Thank you for your attention!

Questions?

Feel free to ask about any aspect of the presentation!

Sources and References I

- MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations." *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281-297.
- Ward Jr., J. H. (1963). "Hierarchical grouping to optimize an objective function." *Journal of the American Statistical Association*, 58(301), 236-244.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.
- Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). "Data clustering: A review." *ACM Computing Surveys*, 31(3), 264-323.
- Kaufman, L., & Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.
- Pedregosa, F., et al. (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12, 2825-2830.

Thank you for your attention!

Md Amanullah (5466324)

Machine Learning - CS2356